

\$index of Object in Array while using ng-repeat and a filter

Im fairly new to angular and have been able to get around somewhat. But I cant seem to find the answer to this scenario...

I have an array of objects, which I am pulling down from firebase. I am using an ng-repeat for the objects, then displaying data accordingly. I am trying to pass the index as a routeparam to an "edit" controller. In which case I would like to pull the object data as one would anticipate. However, when I filter the ng-repeat I am getting the index of the filtered content. where am I going wrong in finding the true index?

```
.config(['$routeProvider', '$locationProvider', function($routeProvider,
$locationProvider) {
$routeProvider
.when('/profiles/:index', {
templateUrl: '../views/profile.html',
controller: 'profileCtrl'
});
});
```

Route is above, Controller is below

```
.controller('profileCtrl', function( $scope, $routeParams ){
$scope.teamProfile = $scope.ourTeam[$routeParams.index];
$scope.index = $routeParams.index;
});
```

And finally the snippet of html from within the repeat.

```
<div class="profileName"><a href="/profiles/{{ $index }}">{{ member.name }}</a><span
class="handle">{{ member.handle }}</span></div>
```

javascript arrays angularjs angularjs-ng-repeat

asked Nov 21 '13 at 3:02

 **lutonmedia**
43 1 1 3

2 Did you mean `{{index}}` instead of `{{ $index }}` ? – [elclanrs](#) Nov 21 '13 at 3:08

5 Answers

Unfortunately `$index` is only the "iterator offset of the repeated element (0..length-1)"

If you want the original index you would have to add that to your collection before filtering, or simply not filter the elements at all.

One possible approach:

```
angular.forEach(members, function(member, index){
//Just add the index to your item
member.index = index;
});

<div ng-repeat="member in members">
  <a href="/profiles/{{member.index}}">
</div>
```

Now, it also seems that this kind of information **is really more like an ID** than anything else. Hopefully that is already part of the record, and you can bind to that instead of using the index.

edited Nov 21 '13 at 3:16

answered Nov 21 '13 at 3:10

 **Josh**
34.3k 5 67 109

and what if you are iterating through array of strings? you will not able to add attribute to it (index). – [vlio20](#)
Jul 20 '15 at 20:55



Find your dream job
on a career site built just for developers



Get started

Try this :

```
<div ng-repeat="member in members">
  {{members.indexOf(member)}}
</div>
```

indexOf always returns the original index in an ng-repeat

Demo

answered May 7 at 7:26



Manas Hemrajani

252 2 9

Why isn't this solution the most accepted - is there something I'm missing as it seems to work perfectly for me? – [userMod2](#) May 8 at 13:56

Its not the most accepted, i think, because its relatively so new. – [Manas Hemrajani](#) May 9 at 6:57

2 this should be the answer. :) – [chitcharonko](#) Jul 8 at 15:58

You could use a function to return the index from the array

```
<div ng-repeat="post in posts | orderBy: '-upvotes'">
  <a href="#/posts/{{getPostIndex(post)}}"></a>
</div>
```

And the function

```
$scope.getPostIndex = function (post) {
  return $scope.posts.indexOf(post); //this will return the index from the array
}
```

On my example I have an array of objects called "posts", on which I use a filter to order them by one of their properties ("upvotes" property). Then, in the "href" attribute I call "getPostIndex" function by passing it **by reference**, the current object.

The getPostIndex() function simply returns the index from the array by using Javascript array indexOf() method.

The nice thing about this is that this solution is not tied to a specific filter (like in [@holographix](#) answer) and will work for all of them.

answered Mar 5 at 22:22



Soky

51 1 3

I just stumbled across the same problem and I found this supertrick on the angular git issues

```
items.length - $index - 1
```

like

```
<div ng-repeat="item in ['item', 'item', 'item'] | reversed">
  <!-- Original index: 2, New index: 0 -->
  <p>Original index: {{items.length - $index - 1}}, New index: {{ $index }}</p>
  <!-- Original index: 1, New index: 1 -->
  <p>Original index: {{items.length - $index - 1}}, New index: {{ $index }}</p>
  <!-- Original index: 0, New index: 2 -->
  <p>Original index: {{items.length - $index - 1}}, New index: {{ $index }}</p>
</div>
```

if you're in trouble like me give it a shot:

<https://github.com/angular/angular.js/issues/4268>

answered Dec 2 '14 at 16:24



holographix

1,320 14 36

If your not using a filter this would be the best solution. The filter would screw up your index. – [Mehdi Abderezai](#) Feb 8 at 20:06

You can inject `$route` and use `$route.current.params.index` to get the value.

```
.controller('profileCtrl', function( $scope, $route ) {  
    $scope.index = $route.current.params.index;  
});
```

edited Jun 25 '14 at 14:15



[Josh Lowe](#)
424 3 13

answered Nov 21 '13 at 3:37



[Awakening](#)
709 1 15 31
